Linear Support Vector Machine

Gilles Gasso

INSA Rouen - ASI Department
Laboratory LITIS

December 8, 2020

Road map

Linear discrimination

Goal

- Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \{-1, 1\}\}_{i=1\cdots n}$: be a set of labeled samples
- Using $\mathcal{D}$, train a classification function $f : \mathcal{X} \rightarrow \{-1, 1\}$ or $f : \mathcal{X} \rightarrow \mathbb{R}$ able to predict the true class of $\boldsymbol{x} \in \mathcal{X}$

Bus

Train

Formulation

- $\mathcal{D} = \{(x_i, y_i) \in \mathcal{X} \times \{-1, 1\}\}_{i=1\cdots n}$: training set

Classification function

- Let the input space be $\mathcal{X} = \mathbb{R}^d$
- Scoring function: $f : \mathbb{R}^d \rightarrow \mathbb{R}$ such that if

$$\begin{array}{ll} f(\boldsymbol{x}) < 0 & \text{assign } \boldsymbol{x} \text{ to class } -1 \\ f(\boldsymbol{x}) > 0 & \text{assign } \boldsymbol{x} \text{ to class } 1 \end{array}$$

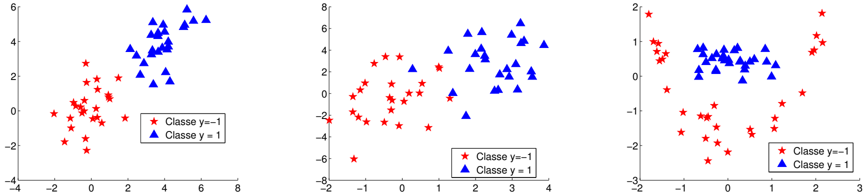
- Linear function:

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b, \qquad \boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}$$

Definition

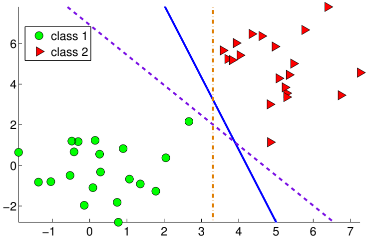
Linearly separable classification problem

The data $\{(\mathbf{x}_i, y_i)\}$ are linearly separable if it exists a separating hyperplane which classifies correctly the samples. Otherwise, the problem is not linearly separable.



2D example

Find a perfect linear classification function of the samples



- Decision function: $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$
- Several solutions exist
- Do these solutions come equally?

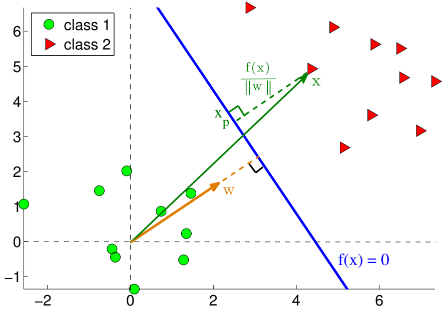
A potential solution to pick up

Select the margin maximizing classification function

Notion of geometry

Distance to the decision boundary

Let $H(\mathbf{w}, b) = \{\mathbf{z} \in \mathbb{R}^d \,|\, f(\mathbf{z}) = \mathbf{w}^\top \mathbf{z} + b = 0\}$ be a hyperplane and $\mathbf{x} \in \mathbb{R}^d$ a point. The distance of $\mathbf{x}$ to the hyperplane H is defined as
$d(\mathbf{x}, H) = \frac{|\mathbf{w}^\top \mathbf{x} + b|}{\|\mathbf{w}\|} = \frac{|f(\mathbf{x})|}{\|\mathbf{w}\|}$



Let $\mathbf{x}_p$ be the orthogonal projection of $\mathbf{x}$ onto H.

We have $\mathbf{x} = \mathbf{x}_p + a\frac{\mathbf{w}}{\|\mathbf{w}\|} \rightarrow a\frac{\mathbf{w}}{\|\mathbf{w}\|} = \mathbf{x} - \mathbf{x}_p$.

The dot product with $\mathbf{w}$ leads to $a\mathbf{w}^\top \frac{\mathbf{w}}{\|\mathbf{w}\|} = \mathbf{w}^\top \mathbf{x} - \mathbf{w}^\top \mathbf{x}_p$.

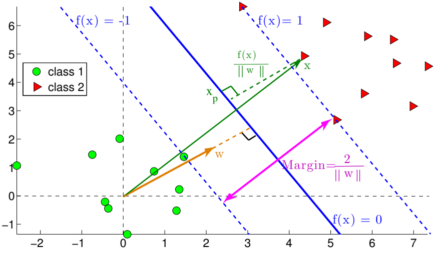
Hence we deduce
$a\frac{\|\mathbf{w}\|^2}{\|\mathbf{w}\|} = \mathbf{w}^\top \mathbf{x} + b - \underbrace{(\mathbf{w}^\top \mathbf{x}_p + b)}_{=0}$.

Therefore we get $a = \frac{\mathbf{w}^\top \mathbf{x} + b}{\|\mathbf{w}\|}$

The margin

Canonical hyperplane

- A hyperplane is canonical w.r.t the data $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$ if $\min_{\boldsymbol{x}_i} |\boldsymbol{w}^\top \boldsymbol{x}_i + b| = 1$



Margin

The geometrical margin is defined as $M = \frac{2}{\|\boldsymbol{w}\|}$

Optimal canonical hyperplane

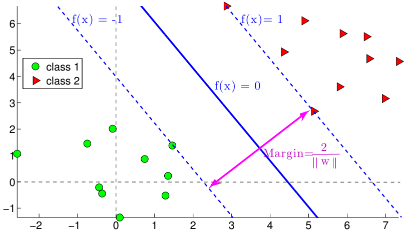
- maximize the margin
- while correctly classifying each sample i.e. $\forall i, \quad y_i f(\boldsymbol{x}_i) > 1$

Maximizing the margin: a formulation

Formulation of SVM

- $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}\}_{i=1}^n$: linearly separable data set
- Goal: determine a function $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$ which maximizes the margin between the classes with no classification error $\mathcal{D}$

$$\begin{array}{lll} \min_{\boldsymbol{w},b} & \frac{1}{2}\|\boldsymbol{w}\|^2 & \text{margin maximization} \\ \text{s.t.} & y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 \quad \forall i = 1, \cdots, n & \text{correct classification} \end{array}$$



The Lagrangian function of SVM problem

Primal

$$\begin{array}{ll} \min_{\boldsymbol{w}\in\mathbb{R}^d, b\in\mathbb{R}} & \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \text{s.t.} & y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 \quad \forall i = 1, \cdots, n \end{array}$$

- Let $\alpha_i \geq 0$, $i = 1 \cdots n$ the Lagrange multipliers related to inequality constraints i.e. n dual variables α_i
- Lagrangian

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1)$$

Dual

- KKT stationary optimality condition

$$\frac{\partial \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial b} = 0 \qquad \frac{\partial \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha})}{\partial \boldsymbol{w}} = 0$$

Soit :

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \qquad \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i$$

- Dual problem: quadratic programming

By substituting the latter relation in $\mathcal{L}$, we atttain:

$$\begin{aligned} \max_{\{\alpha_i\}} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j \\ \text{s.t.} \quad & \alpha_i \geq 0, \quad \forall\, i = 1, \cdots, n \\ & \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned}$$

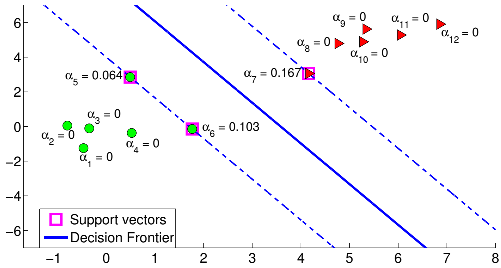
Matrix form

$$\begin{aligned} \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \quad & -\frac{1}{2} \boldsymbol{\alpha}^\top \mathbf{G} \boldsymbol{\alpha} + \mathbf{1}^\top \boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha}, \ \boldsymbol{\alpha}^\top \boldsymbol{y} = 0 \end{aligned}$$

$\mathbf{G} \in \mathbb{R}^{n \times n}$ and $\mathbf{G}_{ij} = y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j$

Support Vectors

- Solve the dual for the parameters $\{\alpha_i\}_{i=1}^n$
- According to the value of α_i we may have the following situations
  - For any sample $\boldsymbol{x}_j$ such that $y_j(\boldsymbol{w}^\top \boldsymbol{x}_j + b) > 1$ we have $\alpha_j = 0$
  - For any $\boldsymbol{x}_i$, if $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) = 1$ then $\longrightarrow \alpha_i \geq 0$
- $\boldsymbol{w} = \sum_{i=1}^n \alpha_i y_i \boldsymbol{x}_i$. $\boldsymbol{w}$ is solely defined on a restricted set of samples such that $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) = 1$. They are called **Support Vectors (SV)**



In practice

Computation of $\boldsymbol{w}$

- Solve the dual using the training set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$

  $\longrightarrow$ We get the dual parameters $\{\alpha_i^*\}_{i=1}^n$
- Obtain the solution as $\boldsymbol{w}^* = \sum_{i=1}^n \alpha_i^* y_i \boldsymbol{x}_i$

Computation of b

- The $\alpha_i^* > 0$ corresponding to the support vectors satisfy the condition

$$y_i({\boldsymbol{w}^*}^\top \boldsymbol{x}_i + b) = 1$$

- Infer b from these relations

Classification function

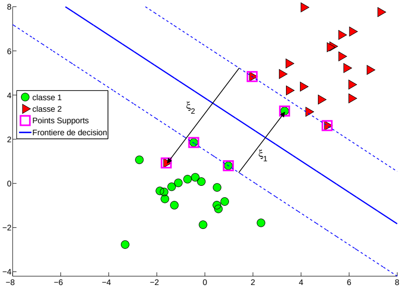
$$f(\boldsymbol{x}) = {\boldsymbol{w}^*}^\top \boldsymbol{x} + b = \sum_{i=1}^n \alpha_i^* y_i \boldsymbol{x}_i^\top \boldsymbol{x} + b$$

Non separable case

What if we cannot find a perfect linear classifier?

Relax the constraints

- Relax $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1$
- and allow $y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i$ with $\xi_i \geq 0$ the slack variables
- Minimize the sum of the slacks $\sum_{i=1}^{n} \xi_i$

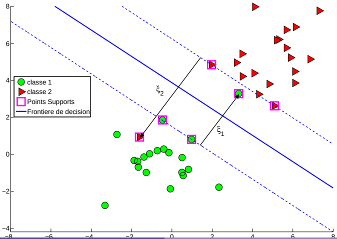


Non separable case: formulation

Linear SVM: general case

$$\begin{array}{lll} \min_{\mathbf{w},b,\{\xi_i\}} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^n \xi_i & \\ \text{s.t.} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i & \forall i = 1, \cdots, n \\ & \xi_i \geq 0 & \forall i = 1, \ldots, n \end{array}$$

- $C > 0$: regularization parameter (controls the trade-off between slack errors and the margin maximization)
- C: selected by the user



Non separable case: dual derivation

Lagrangian

$$\mathcal{L}(\boldsymbol{w}, b, \xi, \alpha, \nu) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^{n}\nu_i\xi_i$$

avec $\alpha_i \geq 0$, $\nu_i \geq 0$, pour tout $i = 1, \cdots, n$

KKT stationary conditions

$$\frac{\partial \mathcal{L}(\boldsymbol{w}, b, \xi_i, \alpha)}{\partial b} = 0 \qquad \frac{\partial \mathcal{L}(\boldsymbol{w}, b, \xi_i, \alpha)}{\partial w} = 0 \qquad \frac{\partial \mathcal{L}(\boldsymbol{w}, b, \xi_i, \alpha)}{\partial \xi_k} = 0$$

give

$$\sum_i^n \alpha_i y_i = 0 \qquad \boldsymbol{w} = \sum_i^n \alpha_i y_i \boldsymbol{x}_i, \qquad C - \alpha_k - \nu_k = 0, \ \forall k = 1 \cdots n$$

The dual problem

Dual

$$\max_{\{\alpha_i\}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^\top x_j$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C, \quad \forall\, i = 1, \cdots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

Matrix form

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^n} \quad -\frac{1}{2} \boldsymbol{\alpha}^\top \mathbf{G} \boldsymbol{\alpha} + \mathbf{1}^\top \boldsymbol{\alpha}$$

$$\text{s.t.} \quad \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}, \ \boldsymbol{\alpha}^\top \boldsymbol{y} = 0$$

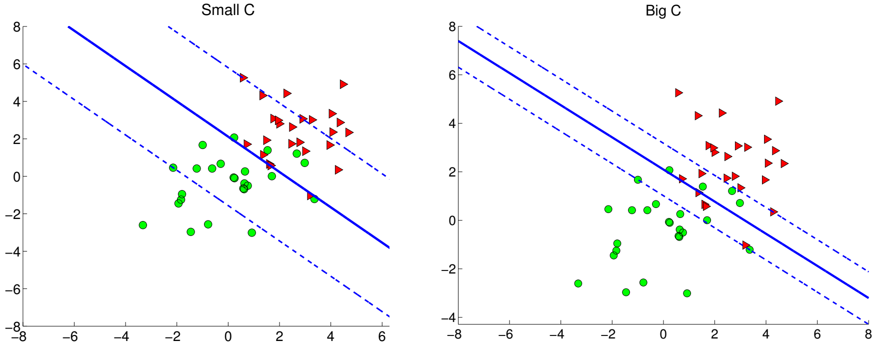
$\mathbf{G} \in \mathbb{R}^{n \times n}$ and $\mathbf{G}_{ij} = y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j$

Computation of $\boldsymbol{w}$

- Given the dual solution $\{\alpha_i^*\}_{i=1}^n$ the SVM parameter vector is given by $\boldsymbol{w}^* = \sum_{i=1}^n \alpha_i^* y_i \boldsymbol{x}_i$
- Compared to linearly separable SVM, the general SVM differs by the box constraints $0 \leq \alpha_i \leq C$ on the α_i.

Influence of the hyper-parameter C

A SVM solved respectively for $C = 0.01$ and $C = 1000$



Influence of C

Small $C \rightarrow$ large margin; large $C \rightarrow$ small margin

Practical Methodology

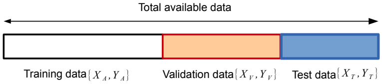
Inputs

Labeled samples : $\{(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}\}_{i=1}^n$

Methodology

1. Center and scale the data : $\{\boldsymbol{x}_i\}_{i=1}^n \longrightarrow \{\boldsymbol{x}_i = \Sigma^{-1}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})\}_{i=1}^n$
2. Fix the hyper-parameter $C > 0$
3. Solve the dual problem to get the $\alpha_i \neq 0$, the corresponding support vector $\boldsymbol{x}_i$ and the bias term b
4. Deduce the classification function $f(\boldsymbol{x}) = \sum_{i \in SV} \alpha_i y_i \boldsymbol{x}_i^\top \boldsymbol{x} + b$
5. Compute the generalization error of the SVM. Repeat from step 2 until a satisfying performance is attained

Tuning C



- Training set: compute $\mathbf{w}$ and b
- Validation set: evaluate the performance of the SVM for different values of C
- Test set: assess the generalization performance of the "best SVM"

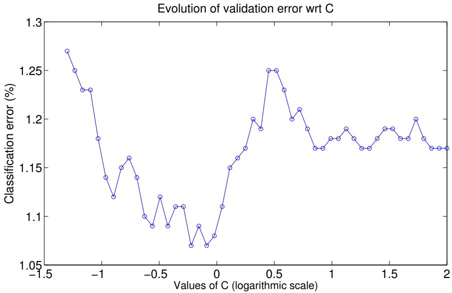
Model selection: tuning C

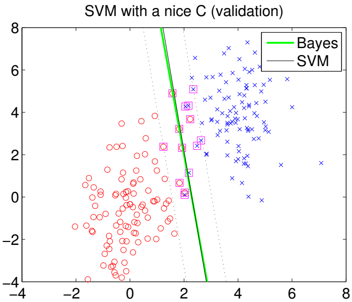
function $C \leftarrow \texttt{tuneC}(\texttt{X}, \texttt{Y}, \texttt{options})$

1. Split the data $(X_a, Y_a, X_v, Y_v) \leftarrow \texttt{SplitData}(\texttt{X}, \texttt{Y}, \texttt{options})$
2. For different values of C
   - $(\mathbf{w}, b) \leftarrow \texttt{TrainLinearSVM}(X_a, Y_a, C, options)$
   - $error \leftarrow \texttt{EvaluateError}(X_v, Y_v, w, b)$
3. $C \leftarrow \arg\min error$

Illustration

- Consider logscale values of C
- For each C value, train an SVM and compute the validation error
- Select the "best SVM" as the minimum of the validation error curve





Multi-class case

K classes $\mathcal{C}_1, \cdots, \mathcal{C}_K$

Common approaches to lift binary SVM to multi-class case:

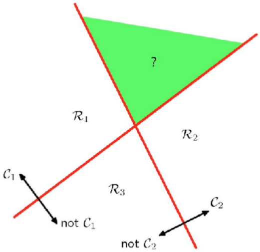
- "One Against All"
  - Learn K SVM (a class against the others)
  - Classify each sample according to the "winner takes all" strategy
- "One Against One"
  - Learn $K(K-1)/2$ SVM (one class against another one)
  - Classify each sample wih a majority vote
  - or estimate the posterior probabilities (pairwise coupling) ; classify according to the maximal posterior probability

Multi-class SVM: One Against All

Dataset : $\{(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \{\mathcal{C}_1, \cdots, \mathcal{C}_K\}\}_{i=1}^N$

Principle

- For each class $\mathcal{C}_k$
    - Learn a binary SVM $f_k(\boldsymbol{x}) = \boldsymbol{w}_k^\top \boldsymbol{x} + b_k$ with data $\{(\boldsymbol{x}_i, z_i) \in \mathbb{R}^d \times \{-1, 1\}\}$
    - where $z_i = 1$ if $y_i = \mathcal{C}_k$ and $z_i = -1$ otherwise



Classifying a new sample $\boldsymbol{x}_\ell$

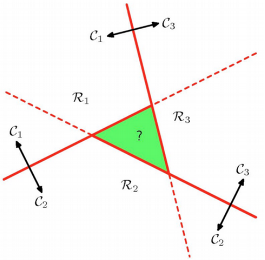
- Winner takes it all
- $D(\boldsymbol{x}_\ell) = \text{argmax}_{k=1,\cdots,K}\{\boldsymbol{w}_1^\top \boldsymbol{x}_\ell + b_1 \cdots, \boldsymbol{w}_k^\top \boldsymbol{x}_\ell + b_k, \cdots, \boldsymbol{w}_K^\top \boldsymbol{x}_\ell + b_K\}$

Multi-class SVM: One Against One

Dataset : $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \{\mathcal{C}_1, \cdots, \mathcal{C}_K\}\}_{i=1}^N$

Principle

- For each pair of classes $(\mathcal{C}_j, \mathcal{C}_k)$
  - Filter out from $\mathcal{D}$ the samples $y_i = \mathcal{C}_j$ or $\mathcal{C}_k$
  - Learn a binary SVM $f_{jk}(\boldsymbol{x}) = \boldsymbol{w}_{jk}^\top \boldsymbol{x} + b_{jk}$ with data $\{(\boldsymbol{x}_i, z_i) \in \mathbb{R}^d \times \{-1, 1\}\}$
  - $z_i = 1$ if $y_i = \mathcal{C}_j$ and $z_i = -1$ if $y_i = \mathcal{C}_k$



Classifying a new sample $\boldsymbol{x}_\ell$: majority vote

- For each learned SVM f_{jk}
  - if $f_{jk}(\boldsymbol{x}_\ell) > 0$ increment the votes for class $\mathcal{C}_j$ otherwise those of $\mathcal{C}_k$
- Assign $\boldsymbol{x}_\ell$ to the class with maximum vote (the one which wins the championship)

To sum up

- Linear SVM for binary classification: maximizes the separation margin between classes while minimizing the classification errors
- Extension to multi-class classification
- Extension to non-linear case using the kernel trick.

Toolboxes
Scikit Learn (Python) implementation
R implementation